

Группа 191-725

Отчет

Объектно-ориентированное программирование

Лабораторная работа №2

ЦИКЛЫ И ВЕТВЛЕНИЯ

Выполнил(а): Бабаев

Проверил(а): Новичков И. К.

2020 г.

# Задача № 1

Предположим, вы хотите создать таблицу умножения на заданное число.

Напишите программу, которая позволяет пользователю ввести это число,

а затем генерирует таблицу размером 20 строк на 10 столбцов. Первые  
строки результата работы программы должны выглядеть примерно следующим образом:

Введите число: 7

7142128354249566370 77849198105112119126133140 147154161168175182189196203210

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(0, "");

unsigned long n;

cout << "Введите число: ";

cin >> n;

for(int j = 1; j <= 200; j++)

{

cout << setw(5) << j \* n << " ";

if(j % 10 == 0)

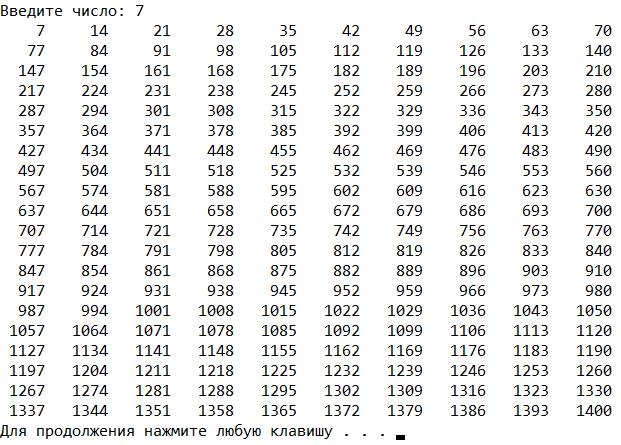
cout << endl;

}

system("pause");

return 0;

}



# Задача № 2

Напишите программу, предлагающую пользователю осуществить

перевод температуры из шкалы Цельсия в шкалу Фаренгейта или

наоборот, а затем осуществите преобразование. Используйте в программе

переменные вещественного типа. Взаимодействие программы с

пользователем может выглядеть следующим образом:

Начните 1 для перевода шкалы Цельсия в шкалу Фаренгейта.

2 для перевода шкалы Фаренгейта в шкалу Цельсия: 1

Введите температуру по Фаренгейту: 70

Значение по Цельсию: 21.111111

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

int response;

double temper;

cout << "1 - для перевода температуры из градусов Фаренгейта в Цельсия,"

<< "\n2 - для перевода температуры из градусов Цельсия в Фаренгейт : ";

cin >> response;

if( response == 1 )

{

cout << "Введите температуру в Фаренгейтах : ";

cin >> temper;

cout << "В градусах Цельсия : " << 5.0/9.0\*(temper-32.0);

}

else

{

cout << "Введите температуру в Цельсиях : ";

cin >> temper;

cout << "В градусах Фаренгейта : " << 9.0/5.0\*temper + 32.0;

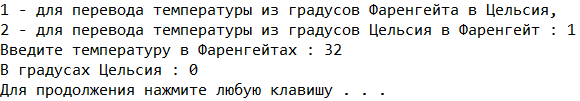
}

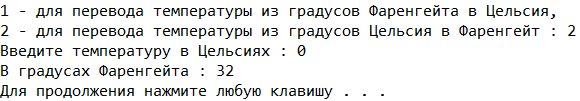
cout << endl;

system("pause");

return 0;

}





# Задача № 3

Операции ввода, такие, как cin, должны уметь преобразовывать

последовательность символов в число. Напишите программу, которая

позволяет пользователю ввести шесть цифр, а затем выводит результат типа

long на экране. Каждая цифра должна считываться отдельно при помощи

функции getche(). Вычисление значения переменной производится путем

умножения текущего ее значения на 10 и сложения с последней введенной цифрой

(для того, чтобы из кода символа получить цифру, вычтите из него 48 или '0').

Примером результата работы программы может служить следующий:

Введите число: 123456

Вы ввел/ число 123456

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

setlocale(0, "");

char ch;

unsigned long total = 0;

cout << "Введите число: ";

while( (ch=getche()) != '\r' )

total = total\*10 + ch-'0';

cout << "\nЧисло: " << total << endl;

system("pause");

return 0;

}



# Задача № 4

Создайте эквивалент калькулятора, выполняющего четыре основных арифметических операции. Программа должна запрашивать ввод пользователем первого операнда, знака операции и второго операнда. Для хранения операндов следует использовать переменные вещественного типа. Выбрать операцию можно при помощи оператора switch. В конце программа должна отображать результат на экране. Результат работы программы с пользователем может выглядеть следующим образом:

Введите первый операнд, операцию и второй операнд: 10 / 3

Результат равен 3.333333

Выполнить еще одну операциию (у/n)? у

Введите первый операнд, операцию и второй операнд: 12 + 100

Результат равен 112

Выполнить еще одну операцию (у/n)? n

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

double n1, n2, ans;

char oper, ch;

do

{

cout << "Введите первое число, потом оператор, затем второй число: ";

cin >> n1 >> oper >> n2;

switch(oper)

{

case '+': ans = n1 + n2; break;

case '-': ans = n1 - n2; break;

case '\*': ans = n1 \* n2; break;

case '/': ans = n1 / n2; break;

default: ans = 0;

}

cout << "Ответ = " << ans;

cout << "\nПроводить операции еще? \nВведите 'y' для - Да\nВведите 'n' для - Нет\n";

cin >> ch;

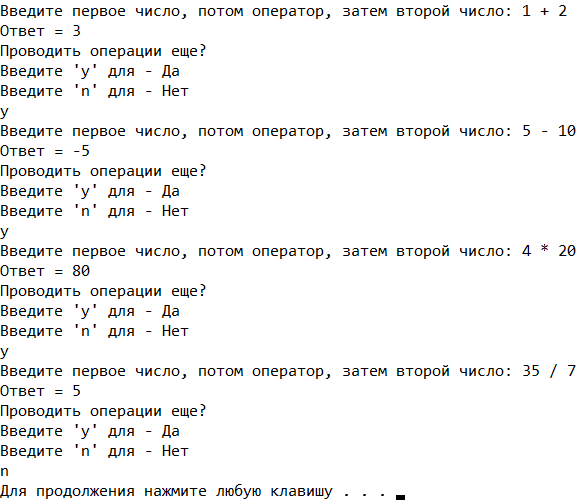
}

while( ch != 'n' );

system("pause");

return 0;

}



# Задача № 5

При помощи цикла for изобразите на экране пирамиду из символов 'X'. Верхняя часть пирамиды должна выглядеть следующим образом:

х

ххх

ххххх

ххххххх

ххххххххх

Вся пирамида должна быть высотой не 5 линий, как изображено здесь, а 20 линий. Одним из способов ее построения может служить использование двух вложенных циклов, из которых внутренний будет заниматься печатанием символов 'X' и пробелов, а другой осуществлять переход на одну строку вниз.

#include <iostream>

#include <string>

using namespace std;

int main()

{

for (int i = 0; i < 20; i++)

{

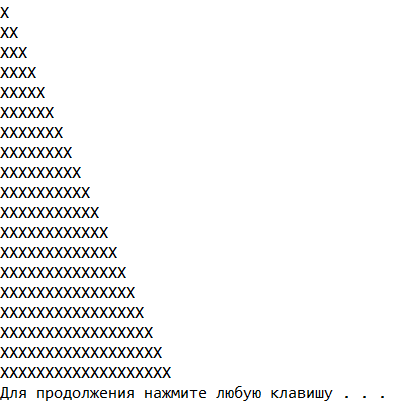
cout << string(i, 'X') << endl;

}

system("pause");

return 0;

}



# Задача № 6

Измените программу factor таким образом, чтобы она циклически запрашивала ввод пользователем числа и вычисляла его факториал, пока пользователь не введет 0. В этом случае программа должна завершиться. При необходимости вы можете использовать соответствующие операторы программы factor в цикле do или while.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

unsigned int n;

unsigned long fact = 1;

do//Для цикличного спроса чисел, и высчитывания факториалов

{

cout << "Введите число: ";

cin >> n;

if (n > 0)

{

for (int j = n; j > 0; j--)//Высчитывает факториал

{

fact \*= j;

}

cout << "Факториал : " << fact << endl;

fact = 1;

}

else return 0;//Условие выхода из цикла, если переменная == ноль

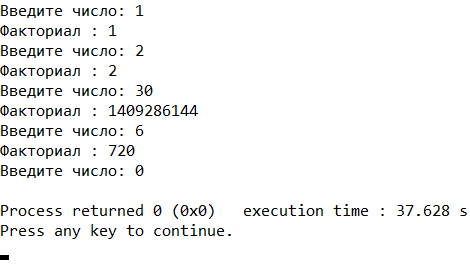
}

while (fact > 0);

system("pause");

return 0;

}



# Задача № 7

Напишите программу, рассчитывающую сумму денег, которые вы получите при вложении начальной суммы с фиксированной процентной ставкой дохода через определенное количество лет. Пользователь должен вводить с клавиатуры начальный вклад, число лет и процентную ставку. Примером результата работы программы может быть следующий:

Введите начальный вклад: 3000

Введите число лет: 10

Введите процентную ставку: 5.5

через 10 пет вы получите 5124.43 доллара.

В конце первого года вы получите 3 000 + (3 000\*0.055) = 3165 долларов. В конце следующего года вы получите 3 165 + (3 165\*0.055) = 3339.08 долларов. Подобные вычисления удобно производить с помощью цикла for.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

unsigned int time;

double per, dep;

cout << "Введите начальный вклад: ";

cin >> dep;

cout << "Введите число лет: ";

cin >> time;

cout << "Введите процентную ставку: ";

cin >> per;

per /= 100;

for (int i = 0; i < time; i++)

{

dep = dep + (dep \* per);

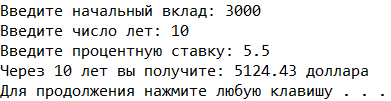
}

cout << "Через " << time << " лет вы получите: " << dep << " доллара" << endl;

system("pause");

return 0;

}



# Задача № 8

Напишите программу, которая циклически будет запрашивать ввод поль  
зователем двух денежных сумм, выраженных в фунтах, шиллингах и пен  
сах (см. упражнения 10 и 12 предыдущего набора упражнений). Программа должна складывать введенные суммы и выводить на экран результат, также выраженный в фунтах, шиллингах и пенсах. После каждой итерации программа должна спрашивать пользователя, желает ли он продолжать работу программы.  
При этом рекомендуется использовать цикл do. Естественной формой  
взаимодействия программы с пользователем была бы следующая:

Введите первую сумму £5 10 6

Введите вторую сумму £3 2 6

Всего £8 13 0

Продолжить (у/n)?

Для того чтобы сложить две суммы, вам необходимо учесть заем одного шиллинга в том случае, если число пенсов окажется больше 11, и одного фунта, если число шиллингов окажется больше 19.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

int fun1, sh1, pen1, fun2, sh2, pen2, fun, sh, pen, sum1, sum2, raz1, raz2;

char ans;

cout << "Для продолжения работы введите на клавиатуре символ (Y)" << endl;

cout << "Для отказа работы введите на клавиатуре символ (N)" << endl;

do

{

cout << "Введите первую сумму : ";

cin >> fun1 >> sh1 >> pen1;

cout << "Введите вторую сумму : ";

cin >> fun2 >> sh2 >> pen2;

sum1 = fun1 \* 240 + sh1 \* 12 + pen1;

sum2 = fun2 \* 240 + sh2 \* 12 + pen2;

sum1 = sum1 + sum2;

fun = sum1 / 240;

fun = static\_cast<int>(fun);

fun1 = fun;

fun = fun \* 240;

raz1 = sum1 - fun;

sh = raz1 / 12;

sh = static\_cast<int>(sh);

raz2 = sh \* 12;

pen = raz1 - raz2;

cout << "Всего: " << fun1 << "." << sh << "." << pen << endl;

cout << "Продолжить? - (Y/N)" << endl;

cin >> ans;

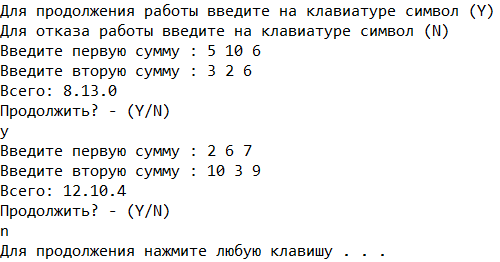
}

while (ans == 'y' || ans == 'Y');

system("pause");

return 0;

}



# Задача № 9

Представьте, что вы собираетесь пригласить к себе шестерых гостей, но за  
вашим столом могут разместиться всего лишь 4 человека Сколькими  
способами можно разместить четырех из шести гостей за обеленным сто  
лом? Каждый из шести гостей может разместиться на первом стуле. Каж  
дый из оставшихся пяти гостей может занять второй стул. На третьем  
стуле может разместиться один их четырех гостей, и на четвертом — один  
из трех оставшихся гостей. Двоим из гостей не достанется ни одного  
места. Таким образом, число возможных рассадок гостей за столом равно  
6\*5\*4\*3 = 360. Напишите программу, которая будет производить аналогич  
ные вычисления для любого числа гостей и любого числа мест за столом  
(при этом предполагается, что число гостей не меньше числа мест). Про  
грамма не должна быть сложной, и вычисление можно организовать с по  
мощью простого цикла for.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

int stul, gost, var = 1;

cout << "Введите количество мест : ";

cin >> stul;

cout << "Введите количество гостей : ";

cin >> gost;

for (int i = 0; i < stul; i++)

{

var = var \* gost;

gost--;

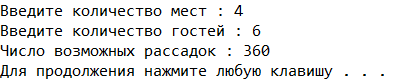
}

cout << "Число возможных рассадок : " << var << endl;

system("pause");

return 0;

}



# Задача № 10

Модифицируйте программу, описанную в упражнении 7, так, чтобы вместо вычисления текущей суммы на вашем счете она вычисляла, сколько лет потребуется для того, чтобы при заданной процентной ставке и величине начального вклада сумма на вашем счете достигла запрашиваемого вами значения. Для хранения найденного числа лет используйте переменную целого типа (можно отбросить дробную часть значения, полученного в результате расчета). Самостоятельно выберите тип цикла, подходящий для решения задачи.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

unsigned int time = 0;

double per, dep, dep1;

cout << "Введите начальный вклад: ";

cin >> dep;

cout << "Введите Итоговую сумму вклада: ";

cin >> dep1;

cout << "Введите процентную ставку: ";

cin >> per;

per /= 100;

while (dep < dep1)

{

dep = dep + (dep \* per);

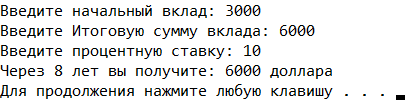
time++;

}

cout << "Через " << time << " лет вы получите: " << dep1 << " доллара" << endl;

system("pause");

return 0;

}

# Задача № 11

Создайте калькулятор, выполняющий действия над денежными суммами, выраженными в фунтах, шиллингах и пенсах (см. упражнения 10 и 12 предыдущего набора заданий). Калькулятор должен складывать и вычитать вводимые значения, а также производить умножение денежной суммы на вещественное число (операция умножения двух денежных сумм не имеет смысла, поскольку квадратных денежных единиц не существует. Деление одной денежной суммы на другую мы тоже не будем рассматривать). Организация взаимодействия с калькулятором описана в упражнении 4 этого набора упражнений.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

int fun1, sh1, pen1, fun2, sh2, pen2, fun, sh, pen, sum1, sum2, raz1, raz2;

char ans, sign;

cout << "Для продолжения работы введите на клавиатуре символ (Y)" << endl;

cout << "Для отказа работы введите на клавиатуре символ (N)" << endl;

cout << "Пример написания: (первое число) (оператор) (второе число)" << endl;

do

{

cout << "Введите первую сумму : ";

cin >> fun1 >> sh1 >> pen1;

cin >> sign;

cout << "Введите вторую сумму : ";

cin >> fun2 >> sh2 >> pen2;

sum1 = fun1 \* 240 + sh1 \* 12 + pen1;

sum2 = fun2 \* 240 + sh2 \* 12 + pen2;

switch (sign)

{

case '+': sum1 = sum1 + sum2;

break;

case '-': sum1 = sum1 - sum2;

break;

default: sum1 = 0;

}

fun = sum1 / 240;

fun = static\_cast<int>(fun);

fun1 = fun;

fun = fun \* 240;

raz1 = sum1 - fun;

sh = raz1 / 12;

sh = static\_cast<int>(sh);

raz2 = sh \* 12;

pen = raz1 - raz2;

cout << "Итог : " << fun1 << "." << sh << "." << pen << endl;

cout << "Продолжить? - (Y/N)" << endl;

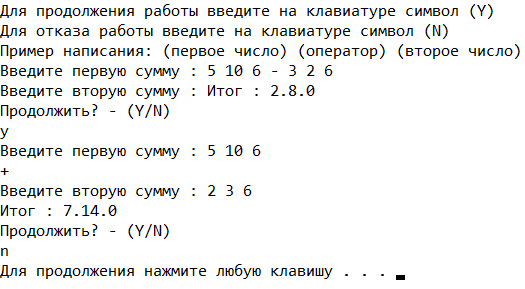
cin >> ans;

}

while (ans != 'n' || ans != 'N');

system("pause");

return 0;

}

# Задача № 12

Создайте калькулятор, выполняющий четыре арифметических действия над дробями (см. упражнение 9 предыдущего набора и упражнение 4 этого набора). Формулы, демонстрирующие выполнение арифметических операций над дробями, приведены ниже.

Сложение: a/b + c/d = (a\*d + b\*c)/(b\*d)

Вычитание: a/b - c/d = (a\*d - b\*c)/(b\*d)

Умножение: a/b\*c/d = (a\*c)/(b\*d)

Деление: a/b/c/d = (a\*d)/(b\*c)

Пользователь должен сначала ввести первый операнд, затем знак операции и второй операнд. После вычисления результата программа должна отобразить его на экране и запросить пользователя о его желании произвести еще одну операцию.

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "");

double a, b, c, d, itog;

char ans, sign, sign1;

cout << "Для продолжения работы введите на клавиатуре символ (Y)" << endl;

cout << "Для отказа работы введите на клавиатуре символ (N)" << endl;

cout << "Пример написания: (первое число) (оператор) (второе число)" << endl;

do

{

cout << "Введите первую дробь : ";

cin >> a >> sign1 >> b;

cin >> sign;

cout << "Введите вторую дробь : ";

cin >> c >> sign1 >> d;

switch (sign)

{

case '+': itog = (a \* d + b \* c) / (b \* d);

break;

case '-': itog = (a \* d - b \* c) / (b \* d);

break;

case '\*': itog = (a \* c) / (b \* d);

break;

case '/': itog = (a \* d) / (b \* c);

break;

default: itog = 0;

}

cout << "Итог : " << itog << endl;

cout << "Продолжить? - (Y/N)" << endl;

cin >> ans;

}

while (ans == 'y' || ans == 'Y');

system("pause");

return 0;

}

